# **Chatbot Report**

## Introduction

This report outlines the steps taken to develop and improve a chatbot using Hugging Face models in Streamlit. It also highlights challenges faced and solutions implemented.

## Steps Taken

1. **Project Setup**
   * Installed necessary dependencies including transformers, torch, and streamlit.
   * Configured a Python environment and initialized a GitHub repository.
2. **Model Selection and Integration**
   * Evaluated different Hugging Face models such as gpt2, gpt-neo, and opt.
   * Loaded tokenizer and model using AutoTokenizer and AutoModelForCausalLM.
3. **Streamlit UI Implementation**
   * Created a simple user interface for chatbot interaction.
   * Used st.text\_input for user queries and st.write for responses.
4. **Testing and Debugging**
   * Ran local tests to verify model outputs.
   * Addressed errors related to missing dependencies and API integration.
5. **Version Control and Deployment**
   * Configured Git for version control.
   * Resolved issues related to GitHub push protection due to exposed API keys.

## Challenges Faced and Solutions

1. **GitHub Push Protection Blocked Commits**
   * Problem: GitHub blocked pushes due to detected secrets.
   * Solution: Removed sensitive data from commit history and used .env files.
2. **Model Loading Issues**
   * Problem: Model downloads were slow or failed due to internet issues.
   * Solution: Cached models locally and verified file paths.
3. **Streamlit Errors**
   * Problem: ScriptRunContext warnings in Streamlit.
   * Solution: Ensured the script was run using streamlit run chatbot.py.

## Improvements and Next Steps

* Implement a more advanced model for better responses.
* Optimize model inference speed.

## Conclusion

This project successfully integrated a chatbot using Hugging Face models and Streamlit. Despite some challenges, the implementation was improved through debugging and security enhancements.

Github Repository: <https://github.com/dayoxy/Module-5.git>

Chatbot Preview



